Capturing Intuitions About Human Language Production

Nigel Ward

University of Tokyo, Faculty of Engineering
University of California at Berkeley, Computer Science Division

Abstract

Human speech is creative. Move specifically, it is an effortful process that starts from a rich input and creates new meaning. Speech is also incremental, as evidenced by pauses and false starts. Existing models of language generation have not adequately addressed these phenomena. This paper presents six principles which specify a design for a cognitively plausible generator, as follows: Be able to handle non-trivial inputs, Be able to access relevant words, Consider many words as candidates for potential inclusion, Produce an utterance incrementally, Use feedback from words, and Monitor the output. These principles can be implemented using spreading activation in a semantic network which includes lexical and syntactic knowledge. The prototype generation system FIG is such an implementation.

Introduction

The study of human language production is still in its infancy. No existing model approaches even descriptive adequacy. Many open problems are well known, such as problems of choosing a referring expression or choosing the correct position for an adjunct. But there are deeper problems, stemming from the failure to address some very basic issues in language production.

[0] Thanks to Dan Jurafsky, Terry Regier, Michael Gasser, Jim Martin, Y. Kuniyoshi, H. Inoue, Robert Wilensky, and Dekai Wu. This work was supported in part by a Monbusho Fellowship, by the Sloan Foundation via a grant to the Berkeley Cognitive Science Program, and by the Defense Advanced Research Projects Agency (DoD), monitored by the Space and Naval Warfare Systems Command under Contract N00039-88-C-0292.

In particular, human speech is creative and incremental. These two phenomena have received little attention, perhaps because they are hard to formalize. But it is worth the effort to explore their implications, for they place important constraints on cognitive models. This paper discusses these basic intuitions, proposes some qualities that a generator needs in order to capture them, and describes a generation program designed to exhibit these phenomena.

Phenomena

Speech is Creative

"Creative" is here a sort of umbrella term for some important intuitions about the nature of speech:

1. The "input" is much richer than the output. In Vygotsky's words, "thought, unlike speech, does not consist of separate units. When I wish to communicate the thought that today I saw a barefoot boy in a blue shirt running down the street, I do not see every item separately: the boy, the shirt, its blue color, his running, the absence of shoes" [Vygotsky 1982]. This points up the computational tasks of dividing and organizing the input.

Rich inputs cause another problem as well, that of selection. Imagine a peach moving in a stream towards a woman standing by the side of the stream. If you have formed a rich image of this scene, then you can think of many words that could be used in a sentence describing it, including perhaps *"water," "flow," "float," "in," "on," "downstream," "down," "towards," "to," "moving," "direction," "buoyant," "bobbing," "current," "with," "stream,"* and so on. No normal utterance will contain all these words, so a speaker must select which aspects of an input to express.

2. Speech creates meaning. Those having difficulty organizing an idea often use the strategy of verbalizing their thoughts, hoping to understand them better. This suggests that speaking involves creating new conceptual content, and is not merely a process of encoding existing thoughts. In Wittgenstein's words, it is a mistake to assume that "the thoughts are already there ... and we merely look for their expression" [Wittgenstein 1963].

3. Speech is effortful. As suggested by the common saying "he was so busy talking that he never noticed ...," even the folk theory of language includes the notion that speech imposes a cognitive load.

Speech is Incremental

Consider the following utterance:

> *Nigel, you've got such a we-ird answering machine message. - so anyway we need to know what time we can come and rehearse tomorrow, - a-nd, - we're assuming that, - ten o'clock in the morning we'll show up at your place, — and, — or or shortly thereafter, - and uh - if we - if we don't hear from you then we'll do that, if you - can't do that then leave a message on my machine ok? - thanks bye.*

Notation:

- \- short pause or lengthened word
- — longer pause
- ,.? clause boundaries, based on intonation

Now that you understand the content, a second reading will reveal:

1. Speech is full of *pauses,* including filled pauses such as *"uh"* and *"oh."* The obvious and pervasive explanation for this is that people think as they speak; in particular, that "deciding" what to say next requires thought. In other words, the "effort" of speaking is expended "on-line."

2. Speech exhibits *false starts,* that is, cases where the speaker utters a few words, "denies" them, and starts over. The obvious explanation for this is that speakers choose words without completely computing the consequences of what they are likely to say next. In Kempen's words, "speakers often ... initiate overt speech production after having worked out only a fragment of the conceptual content of the resulting utterance" [Kempen & Hoenkamp 1987].

Speech is Successful

For the sake of completeness it should be noted that people usually manage to say what they mean, and they do so more or less according to the rules of their language. Obeying the rules means producing an utterance which is not only grammatical but also consists of a consistent set of words. The complexity involved in being consistent can be seen by considering the many ways in which a preposition can be inconsistent with the other words in an utterance:

> *went* into *the hills to gather wood*
> **went* into *the hill to gather wood*
> (but *"to"* is ok)
> **went* into *the hillside to gather wood*
> (but *"gather wood* on *the hillside"* is ok)
> **?walked into the hills to gather wood*

Thus, the choice of a preposition can depend on a previous word, a latter word, the number of that word, and so on.

Trends in Generation Research

Human language production has received a fair amount of attention, but the phenomena of creativity and incrementality have been largely neglected. This section points out some broad trends in generation research and discusses the extent to which the results are plausible cognitive models. The strengths and weaknesses of specific models are discussed in later sections.

Psycholinguists' models of generation have not addressed creativity, nor, in general, incrementality. Instead, they generally target quantifiable data, such as pause durations and rates, speech errors, and response latencies. These are typically explained directly in terms of a speech "machine" and its phases, levels, modules, or stages (for a survey of many such models, see [Rosenberg 1977]). In general, much of this work seems prematurely concerned with details at the expense of the big picture, and also too willing to accept the Von Neuman computer as a model for cognition.

In the AI tradition "cognitive validity" usually refers to the use of data *structures* which plausibly reflect human knowledge. There has been less use of intuitions about *processing* aspects of language skills. There are exceptions, notably Kempen's work in incremental generation [Kempen & Hoenkamp 1987]. But even this seems mostly motivated by the goal of producing syntacticly correct output. The result is yet another tree-based syntactic engine, albeit one capable of generating even while the speaker's intention is changing.

Most researchers are not terribly concerned if their models cannot account for creativity. Generating successfully is usually given first priority. A typical viewpoint is "while there are creative aspects in speech, they should be handled separately from the basic generation process." Yet it seems wrong not to try for a unified model that can account for both creative aspects and more mundane aspects of generation.

Indeed, it may be that creativity and incrementality are not merely incidental phenomena but are intrinsic to the generation task: to the extent that thought is creative and unconstrained, so must language be; to the extent that language is concise, it must reflect a creative selection of elements from a thought; to the extent that speech is produced in real time and by a speaker situated in a changing world, it must be produced incrementally.

Principles for a Cognitive Model of Language Production

The following principles describe how a generator should operate if it is to be cognitively plausible. Taken together, they form a kind of "specification" for a generator.

Principle 1: Be Able to Handle Non-Trivial Inputs

A lot of things are "active in the mind" and affect the generation process, as illustrated by Vygotsky's example and by the telephone message. Pragmatic goals are also active in large numbers; in Hovy's words, "when we speak, we do not try to satisfy only one or two goals, and we operate (often, and with success) under conflicting goals for which no resolution exists" [Hovy 1987]. Indeed, generation can be affected by utterly extraneous thoughts [Harley 1984]. It may not be too much to say that the generator's input is the entire brain-state of the speaker. For this reason to use the word "input" is somewhat inappropriate, in so far as it evokes the image of a generation process running as an isolated module that is passed a few symbols and has no access to other components of thought.

Yet this is exactly the image that many generation researchers have adopted. They assume the "input" can be adequately modeled as a logical form, feature structure, conceptualization, set of propositions, or the like. As a consequence, the most widely used metaphor for generation, even in psychological circles, seems to be that of encoding a thought or a "meaning." The computational analog of this is to treat generation as a process of mapping one representation (a thought) onto another (a sentence). This is typically done by a matching or unification process or by a process that "traverses" the input structure. This metaphor assumes, and its implementations require, a simple, well-organized and contextless input, which makes them implausible as cognitive models.

Another common metaphor for generation is realizing a specification. This metaphor seems better suited for explaining the ability to handle non-trivial inputs. Yet the computational models based on this metaphor are also unsatisfactory. The first, syntax-driven generation, seems hard to reconcile with the creative, meaning-creating aspects of generation. The second is planning-based generation. Planning techniques seem, unfortunately, too general to cast much light on the specific task of language production.

Principle 2: Be Able to Access Relevant Words

In order to "say what it means" a generator must find words relevant to its input. Relevance is not always a straightforward relationship; many words may qualify as relevant [Ward 1988]. For example, to produce an utterance expressing something like *"Chang lived with his widowed mother"* a generator may need access to words like *"alone," "father," "without," "dead,"* and *"orphan."* This flexibility, sometimes called paraphrase ability, is one key to creativity.

A wealth of possible choices, however, is something that existing generation models (with the exception of planning and unification approaches) cannot cope with. This is because their basic algorithms are fundamentally serial. The problem of producing a grammatical, consistent, and appropriate utterance despite a wealth of choices is the primary topic of the remaining principles.

Principle 3: Consider Many Words as Candidates for Potential Inclusion

Recall the problem of making a consistent set of choices: a word choice which seems excellent in isolation may turn out to be inappropriate when the possibilities for neighboring words are considered. In order to have a good chance of finding a set of choices which are all individually reasonable and together form a consistent set, a generator must consider many alternative words. The previous principle says that a generator should have access to many words; this one says that it should actively consider them "at run time." That is, it should consider many alternative words in parallel. This allows it to represent all dependencies and "solve" them together.

Consideration of many possible words is necessary for another reason also: the huge number of available concepts if the input is non-trivial. A generator will typically directly express only a few of the possible concepts, as the *"floating peach"* example shows. Not until words are considered can a generator know which concepts it should select. Danlos has pointed out that, in general, "conceptual and linguistic decisions ... are all dependent on one another" [Danlos 1984]. (The distinction between strategy and tactics in generation [Thompson 1977] is an attempt to define this problem away.) In most generators, however, the input consists of a small set of propositions or nodes, in which case generation is a trivial matter of mapping each element or literal to an appropriate word.

Most models of generation do not include consider candidate words in parallel. On the contrary, it is common to treat word choice in terms of dictionary lookup, pointer following, or checklist evaluation, all of which generate only a few candidates. Of course, the lookup process is usually constrained by previous choices, but never by probable future choices. Alternatives are usually not considered unless a failure is detected, in which case special processing is started, such as backtrack or a search process to find another word.

It is worth noting that the goal of producing a consistent set of choices may conflict with the goal of using relevant words. That is, a speaker must sometimes make trade-offs between being faithful to his intention and sounding natural and fluent. To make these diverse goals be comeasurate a generator needs some "scoring" mechanism to enable it to determine what set of choices is the best compromise.

Principle 4: Produce an Utterance Incrementally

A cognitive model of generation should be incremental, or in other words, "on-line." That is, most of the processing relating to the output of a word should be done near the time when that word is output. Incremental generation stands in contrast to multi-pass generation and to top-down generation. Multi-pass generation is obviously useless for cognitive modeling — if speakers first planned the content then the syntactic form and then the words, and only then began to speak, there would be no explanation for false starts. Top-down generation also cannot serve as a cognitive model — a process that expands S to a surface string, preserving grammaticality at each step, can hardly account for disfluencies and false starts.

An incremental generator needs an explicit representation of the current state of the generation process at each moment. Most models of generation do without, maintaining state in pointers to sub-trees or lexical entries, or implicitly in the process state. An explicit representation allows straightforward explanation of errors. For example, false starts can be explained in terms of what is under consideration at each point in time, rather than in terms of buffer properties or architectural shortcomings.

There is a superficial contradiction between choosing words one by one and choosing a consistent set of word. There is no contradiction if a generator represents relevance in general, not just for the next output position. For example, if a generator has a representation of the fact that *"the hills"* is likely to emitted sometime soon, it can use this fact to infer that *"into"* is likely to be appropriate now. This, the need to represent future possible choices and their likelihoods, of course requires yet more parallelism.

Principle 5: Use Feedback from Words

"Speech creates meaning" in part because the availability of words during generation affects the thought of the speaker. More concretely, the retrieval of a word, for whatever reason, provides an opportunity to use the associated concept in thought. This is of course a weak version of the Sapir-Whorf hypothesis.

The incrementality principle and the feedback principle together have an interesting implication. Since most of the processing of a word is done just before that word is output, if feedback is present, then concepts associated with that word will become the "focus of processing." This accounts neatly for Chafe's idea of a "moving focus of attention" which successively "lights up" various parts of the input during the course of generation [Chafe 1980].

If the incrementality principle is taken to the extreme it leads to the claim that "there is nothing more to generation than successive choices of words," and that the word order, syntactic structure, and conceptual content of utterances are emergent. In other words, syntax and content are correctly formed without benefit of a mechanism to explicitly make syntactic or conceptual choices. On this view, meaning arises as side effect of the process of finding words.

Principle 6: Monitor the Output

A generator should exploit feedback from past word choices. This is especially important in an incremental generator because it is necessary to make sure that the generation process stays "on track." One thing feedback must ensure is that the system has a correct representation of what fraction of the input has been conveyed and what remains to be said. For example, after emitting *"there was a peach floating in the stream,"* it must realize that information has been conveyed implicitly; this prevents it from continuing redundantly with *"moving downstream"* or *"being moved by the current."* Feedback must also ensure a correct representation of the current syntactic state.

Feedback is necessary because the other principles require the generator to simultaneously satisfy many goals and subgoals. The generator must "want" to express all the input, to be concise, and to be grammatical. To the extent that these goals are all active at run time there must be processes to check and update the current status (satisfied or not) of each goal.

While monitoring is widely advocated, it is not actually very critical for conventional models of generation. For example, direct replacement generators produce an utterance isomorphic to the meaning structure they are given, and hence allow no possibility for creative word choice. Thus the output is guaranteed to (trivially) conform to the input — monitoring is not really needed. Similarly, syntax-driven generators never output a word without knowing in advance the implications for the sentence's syntactic structure. Everything is predictable so no monitoring is needed.

Implementation

Generating with Spreading Activation

The spreading activation framework is well suited for implementing the above principles. The basic ideas behind spreading activation are: a network represents knowledge, the activation levels of nodes in the network represent the process state, and flow of activation among nodes represents evidence. In generation, a node's activation level represents the degree to which it is relevant to the input or to the utterance under construction. Generation being an incremental process, the activation level of a node represents its relevance at the current time. A spreading activation model can embody the principles as follows:

1. The input is a (potentially large) set of activated nodes.

2. Activation flows in all directions, but in particular it flows from concepts to words, via paths comprised of links representing world knowledge and lexical knowledge. Activation flow across long paths corresponds to the chains of inference in order to "access" words that appear in other models. There is no need to separate lexical information into a special data structure; words are, like everything else, simply nodes in the net.

3. At any given time, many nodes have activation, including many which represent words. The network is designed so that its stable states represent consistent sets of choices. In order for it to settle in this way it has links among words that cooccur, and between compatible choices in general.

This basic mechanism is extended to handle syntax. This solution is based on Construction Grammar [Fillmore, Kay, & O'Connor, to appear], a framework in which the basic units of grammatical knowledge are constructions, such as the subject-predicate construction, the noun-phrase construction, the existential-there construction, and various comparative constructions. Constructions are encoded in the network, where their primary role is to transmit activation to words. At run time many constructions have activation, and they co-operate and compete, "trying" to influence which words get chosen. (This conforms nicely to analyses of speech error data which suggest that even normal speech is the result of competing "plans"[Baars 1980].) If the network is designed right, the states it settles into will represent sets of choices which are consistent from the syntactic point of view also.

It is very hard to prove that the output of this model will be grammatical, due to the massive parallelism. One can say that it strives for grammaticality, without being certain of acheiving it. This seems appropriate, since neither is human speech consistently grammatical. Despite this, other models of generation usually give special status to syntactic considerations. For example, syntax is often treated as a set of constraints, or a filter, or an engine. Other construction-based approaches employ some unification or matching process to arbitrate among or coordinate constructions, rather than having them compete at run time.

4. An utterance is produced by periodically selecting and emitting the most highly activated word-node. These are the only explicit choices; all other choices are emergent. Syntactic considerations, for example, manifest themselves only through their effects on the relative activation levels of nodes.

As a result this is a very open architecture. The generator doesn't know or care if extraneous considerations recently altered the activation level of some node. Moreover, it is robust in the face of such changes, that is, a reasonable output should always result, provided only that the network is given time to settle. Thus, like Kempen's IPG [Kempen & Hoenkamp 1987], it can operate in face of changing thoughts, since, no matter what, it just chooses the next word based on activation levels.

5. Activation flows in all directions, and in particular from words to concepts, thereby providing feedback. Since there is a unified knowledge representation, feedback is easy and pervasive.

6. A monitoring process updates activation levels after each word is emitted. This to ensure that activation levels always represent the current state of the generation process. Some of the specific things involved here are: updating the nodes representing the current syntactic state, zeroing out the activation of a word after it has been emitted (to ensure that it will not be selected again immediately), and zeroing out the activation of those nodes of the input which have been expressed.

FIG – A Flexible Incremental Generator

The above scheme has been implemented as a program called FIG (for "Flexible Incremental Generator"). It works in the domain of fairy tales, for no good reason. FIG's task is generating English in the context of machine translation. The above principles are, of course, most appropriate to spontaneous language production, but they are also relevant to machine translation. The ability to produce language is the most important component skill of the translator's art [Seleskovitch 1968]; and creativity, in particular, is necessary in order to produce natural translations [Ward 1989]. One advantage of studying generation in the context of translation makes it harder to "cheat." There is a temptation in generation research to start from inputs which are really just disguised English sentences. FIG generates directly from the output of a Japanese parser/understander, avoiding this temptation.

In connectionist terminology, FIG is a "localist" system. That is, there is a one-to-one correspondence between concepts and nodes. For example, **old-woman**, **long-ago**, **stream**, and **gather** are all nodes, as are words, such as **"woman"**. While systems with distributed representations have many advantages, localist systems are easier to implement, debug, and explain, and have most of the same characteristics.

The purpose of implementation is to identify the functionality needed for generation and to uncover ways to provide it using spreading activation. I make no claims for FIG as a program; I have little confidence in the details of representation (for example, the ontology of link types, or the heuristics for assigning link weights) or the details of processing (for example, the criteria to determine when the network has settled into a stable state). Experiments with FIG, however, are leading to an understanding of what kinds of knowledge are needed for generation and how they should interact. Some of the results on lexical knowledge and word choice are reported in [Ward 1988].

The remainder of this section gives some details to illustrate the FIG approach. Discussion focuses on syntax because it is not obvious how to handle it in a spreading activation framework.

A construction is represented as a node linked to nodes representing semantic and pragmatic contexts where it is appropriate, the syntactic environments in which it can appear, and its internal structure. For example, consider the existential-there construction [Lakoff 1987]. In the net there is the node **ex-there**, and from it links to nodes representing its constituents: **ex-there.first**, **ex-there.second**, and **ex-there.third**. From these constituents are links to, respectively, the word **"there"**, the category **stative-verb**, and the category **noun**. **ex-there** is also linked to **introductory-sentence**, representing that it is appropriate to use this construction when introducing a new person into the domain of discourse.

The current syntactic state is represented by the activation levels of the constituents of constructions. There is parallelism within constructions: "future" constituents all have some degree of activation, proportional to their imminence. (This is necessary to handle backwards dependencies, as in the *"into the hills"* example, and also to handle optional constituents and adjuncts.)

To see how this knowledge is used during generation, suppose that FIG is given an input including concept-nodes like **woman, old, live, poor, fairy-tale, introductory-sentence**, and **hovel**. Suppose further that FIG has already emitted *"once upon a time."* In this context **ex-there** receives activation from **woman**, indirectly, via a path consisting of **"woman"** (the word, not the concept), **noun,** and **ex-there.second**. It also receives activation from **introductory-sentence**. As a result **ex-there** becomes highly activated (relative to other constructions, such as **subject-predicate**, the construction responsible for normal SV order). A large fraction of this activation flows from **ex-there** via **ex-there.first** to **"there"** (since initially the first constituent of every construction is the most highly activated). As a result of this activation, **"there"** becomes the most highly activated of the nodes representing words, so *"there"* will be emitted next.

After *"there"* is output, activation levels are updated to represent the new syntactic situation. As a result **ex-there**'s second constituent, **ex-there.second**, is highly activated. Activation flows from **ex-there.second**, to the feature **stative-verb**, and from there to stative verbs. At this point the word **"live"** will have the highest activation of any word, because it receives activation from both syntactic sources (**stative-verb**) and semantic sources (the node **live**, a component of the input). Thus FIG will emit *"lived"* next. (There is a kludge for morphology).

This example illustrates several things: Activation flow from constituents to words helps determine which word gets chosen and emitted next. There is "bottom-up" activation which flows from words "up" to constructions they could be constituents for. Appropriate words are selected due to a sort of "cooperation" between syntactic and semantic considerations, since the activation level of a word is given by the sum of the amounts of activation received from all sources.

FIG's knowledge network currently includes about 100 nodes and 300 links. It produces utterances like *"Once upon a time there lived an old man and an old woman,"* and *"One day the old man went to the hills to gather wood."* To produce the latter takes about 21 seconds on a Sun 4/110.

Related Work

FIG builds on other connectionist and activation-based models of generation. Dell [1986] developed a connectionist model of the phonetic realization of words. He emphasized the role of bottom-up feedback (in his case from phonemes to words), and realized that activation levels should represent relevance in general, as well as appropriateness for the very next output position. Stemberger [1985] proposed a similar model for lexical access and word order. Gasser [1988] implemented such a model, CHIE, and noted numerous advantages, including robust word choice. Compared with CHIE, FIG is less top-down, more incremental, and relies more on parallelism and emergents.

Directions

While FIG implements the principles for a cognitive model for language production, it does not fully capture the intuitions about language production. It is incremental, often successful, but never yet creative. To make FIG become more successful, I am gradually extending its knowledge network. On the agenda are investigations of how to handle specific topics, such as morphology, phonology, agreement, and ellipsis, within the spreading activation framework. To make FIG exhibit creativity will be harder. This will require massively scaling up the network so that there are an abundance of choices. One difficulty will be that it takes a fair amount of computer time to do experiments, even for small networks. Another is that it takes a lot of effort to extend the network; at some point a learning algorithm will be needed.

References

[Baars 1980] Baars, Bernard J., The Competing Plans Hypothesis: an heuristic viewpoint on the causes of errors in speech, in *Temporal Variables in Speech,* Hans W. Dechert and Manfred Raupach (eds.), Mouton, 1980.

[Chafe 1980] Chafe, Wallace L., The Deployment of Consciousness in the Production of a Narrative, in *The Pear Stories,* Wallace L. Chafe (ed.), Ablex, 1980.

[Danlos 1984] Danlos, Laurence, Conceptual and Linguistic Decisions in Generation, *Coling-84*, 1984.

[Dell 1986] Dell, Gary, A Spreading-Activation Theory of Retrieval in Sentence Production, *Psychological Review* 93, 283-321, 1986.

[Fillmore, Kay, & O'Connor, to appear] Fillmore, Charles J., Paul Kay, and M. C. O'Connor, Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone, *Language,* to appear.

[Gasser 1988] Gasser, Micheal, A Connectionist Model of Sentence Generation in a First and Second Language, PhD Thesis, also Technical Report UCLA-AI-88-13, Computer Science Department, University of California, Los Angeles, 1988.

[Harley 1984] Harley, Trevor, A Critique of Top-down Independent Levels of Speech Production: Evidence from Non-Plan-Internal Speech Errors, *Cognitive Science*, 8, 1984.

[Hovy 1987] Hovy, Eduard, Generating Natural Language Under Pragmatic Constraints, PhD Thesis, also Technical Report YaleU/CSD/RR #521, 1987.

[Kempen & Hoenkamp 1987] Kempen, Gerard, and Edward Hoenkamp, An Incremental Procedural Grammar for Sentence Formulation, *Cognitive Science*, 11, 201-258, 1987.

[Lakoff 1987] Lakoff, George, *Women, Fire, and Dangerous Things*, University of Chicago Press, 1987.

[Rosenberg 1977] Rosenberg, Sheldon, Semantic Constraints on Sentence Production: An Experimental Approach, in *Sentence Production,* Sheldon Rosenberg (ed.), 195-228, Erlbaum, 1977.

[Seleskovitch 1968] Seleskovitch, Danica, *L'intèrprete dans les conférences internationales,* Minard, Paris, 1968.

[Stemberger 1985] Stemberger, Joseph Paul, An Interactive Activation Model of Language Production, in *Progress in the Psychology of Language, Volume 1*, Andrew W. Ellis (ed.), Erlbaum, 1985.

[Thompson 1977] Thompson, Henry, Strategy and Tactics: A Model for Language Production, *Proceedings of the Chicago Linguistics Society*, 13, 1977.

[Vygotsky 1982] Vygotsky, Lev S., *Thought and Language* MIT and Wiley, 1962

[Ward 1988] Ward, Nigel, Issues in Word Choice, *Proceedings of Coling-88*, Budapest, 1988

[Ward 1989] Ward, Nigel, A Model for Natural Machine Translation, *Jouhou Shori Gakkai Dai38kai Zenkoku Taikai*, Tokyo, 1989.

[Wittgenstein 1963] Wittgenstein, Ludwig, *Philosophical Investigations,* Blackwell, Oxford 1963.